Platform Engineer Technical Questionnaire

Hello,

The following are technical questions for you to answer on your own time, we use this to get a sense of your technical and writing skills on a range of topics. You can return a document with your answers after each question (you can provide external links also, this will be relevant for the coding exercise).

Good luck!

1. Curling wetransfer.com

I open a terminal and type in `curl wetransfer.com`. Please describe in as much detail as you wish what happens between the time you press `Enter` and the time you see curl's response.

Here we're looking for an understanding of what happens on the shell and in the kernel, what internet protocols are involved and how they work. We prefer that you show evidence of your understanding with tools rather than internet research.

2. Implement a circuit Breaker

You are tasked with writing a library that implements the **circuit breaker** pattern to make HTTP clients more resilient to failure, e.g. 500 error or timeouts.

Please implement a small library that wraps and invokes an HTTP client; the HTTP client can be mocked out for this implementation. The library should take in 2 inputs: a time window (in seconds) and an error threshold. If the number of errors from the wrapped HTTP client exceeds the error threshold, the library should prematurely return an error, e.g. “CircuitOpenError”. Whenever the number of errors falls below the threshold, the circuit should be closed and requests should flow freely.

You may think to implement error-counting with a simple scalar. Doing so would be a good first iteration. If you have the time, we’d like you to implement a rolling window with an appropriate backing data structure. The backing data structure used to bookkeep failures within a time window is crucial to an efficient solution. However, your solution does not need to be optimal. Please demonstrate a working solution with a test case or by printing to stdout.

This exercise should take no more than 45 minutes of coding time + time for a little polish. Please push your solution to GitHub or Gitlab for us to review!

```
class CircuitBreaker:
  def __init__(self, http_client, error_threshold, time_window):
    self.http_client = http_client
    self.error_threshold = error_threshold
    self.time_window = time_window

  def do_request(self, url):
    # Go nuts!

if __name__ == "__main__":
  # TODO: Instantiate and configure stub_client

  breaker = CircuitBreaker(stub_client, x, y)

  # TODO:
  #
  # Loop N number of times to simulate consecutive requests.
  # Demonstrate that the circuit opens after some number of
  # failures.
  #
  # Then, wait a while; you may sleep().
  #
  # Demonstrate that the circuit has closed and that requests to the
  # stub client flow freely.
```